Published in final edited form as:

IEEE/ACM Trans Comput Biol Bioinform. 2015 ; 12(3): 632–643. doi:10.1109/TCBB.2014.2372783.

Extraction of Individual Filaments from 2D Confocal Microscopy Images of Flat Cells

Saurav Basu [Member, IEEE],
Center for Bioimage Informatics, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

Chi Liu, and
Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

Gustavo Kunde Rohde [Member, IEEE]
Center for Bioimage Informatics and Department of Biomedical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

Abstract

A crucial step in understanding the architecture of cells and tissues from microscopy images, and consequently explain important biological events such as wound healing and cancer metastases, is the complete extraction and enumeration of individual filaments from the cellular cytoskeletal network. Current efforts at quantitative estimation of filament length distribution, architecture and orientation from microscopy images are predominantly limited to visual estimation and indirect experimental inference. Here we demonstrate the application of a new algorithm to reliably estimate centerlines of biological filament bundles and extract individual filaments from the centerlines by systematically disambiguating filament intersections. We utilize a filament enhancement step followed by reverse diffusion based filament localization and an integer programming based set combination to systematically extract accurate filaments automatically from microscopy images. Experiments on simulated and real confocal microscope images of flat cells (2D images) show efficacy of the new method.

Index Terms

Biological filament networks; local network topology; centerline localization; filament extraction

I. Introduction

Filaments with complex network topology are observed in a wide variety of biological systems, most notably the cytoskeletal filaments and the extracellular matrix. The persistence lengths of these filaments are typically in the range of μm to mm that are considerably greater than the resolution of fluorescent imaging (200 nm) and form complex networks. They are fundamental for several biological processes such as mechanical integrity, force generation, cytokinesis and motility. In addition, the cytoskeleton contributes to a wide range of cellular mechanisms such as intracellular signaling and differentiation [2]. Several microscopy modalities such as the widefield, confocal, and total internal reflectance fluorescence (TIRF) microscopy have been extensively used in recent years to image actin with the purpose of investigating the result of external perturbations on the structural

alterations of cytoskletal networks. Several architectural parameters of biological networks such as filament lengths [13], [26], persistence length [19], local orientation [11], [20], [27], [31], curvature distribution and local filament organization [3], when accurately estimated, might shed crucial light on the role of network geometry in cellular and tissue functions.

A. Burgeoning Interest in Enumerating Filament Networks from Flat Cells

The active regions of interest in cytoskeletal dynamics, specially actin networks, in a wide area of investigation in molecular pathology related to cells are confined to within a few hundred nanometers of the basal region of the cell [10]. Most prominently, cellular force generation occurs through basal actin stress fibers ([8], [24]) and cell motility occurs by polymerization at the leading edge of a thin pseudopod ([4]). As such, even with the acquisition of 3D confocal microscopy image stacks, a majority of actin filament dynamics are confined to within a few 2D slices, and most actin network measurements are essentially carried out in to pseudo-2D mode ([11], [20], [27], [31]). This is so because of the aforementioned difficulties in acquiring high resolution 3D fluorescence image data (time, toxicity and photobleaching) as well as the fact that the resolution in the direction perpendicular to the imaging plane is much lower given the 3D point spread function of confocal microscopes. It is therefore a critical achievement in computational post-processing to develop algorithms to measure cytoskeletal filament networks in these pseudo-2D modes, and we examine 2D projections of flat cells from 3D confocal image stacks.

Considerable advances in capabilities of microscopy devices for visualizing biological structures, including hardware and computational post-processing, have resulted in imaging resolutions at nanometer length scales [9], [25]. Unfortunaltely, optical blurring, noise, clutter, as well as the geometric complexity of such dense networks inside cells severly confound quantitative analysis of cellular filaments *in situ* by fluorescence. Consequently, the process of identifying filament distributions from microscopy images is largely qualitative for most experimental scientists.

The complete mathematical enumeration of a biological filament network, or for that matter, any filament network observable in a 2D image remains an unanswered problem of image analysis. Partial solutions such as local orientation [11], [20], [27], [31] and total filament length [13] have been proposed in the recent past. However, a successful methodology that (1) accurately localizes centerlines of individual filaments in a dense network despite the aforementioned confounding factors and (2) extracts individual filaments with a systematic disambiguation of filament intersections and bifurcations is difficult to locate in literature.

Local thresholding methodologies [5] or filament enhancement schemes followed by some sort of binary thinning [15], for example, are often inadequate solutions to for accurate localization since optimal thresholding parameters are difficult to obtain, while binary thinning ignores the intensity profile of the enhanced images (hence causing the errors delineated in [23]. When it comes to extraction of individual filaments (and subsequently computing subsequent network characteristics such as the filament length distribution) is difficult due to the fact that filament centerlines, in a dense environment, naturally intersect and bifurcate. Greedy algorithms such as following graph edges of a suitably constructed centerline graph (for example, by binarizing and thinning the converged centerline particles

from our previous step) generally give very inaccurate results due to the noise in the graph nodes and ambiguity of multiple possible routes at a graph bifurcation node. It should be noted that some research exists on tracing actin and microtubule filaments from 3D electron tomography images [21], [22], [30], but the very high resolution needed for the point based 3D tracking to be successful is a missing ingredient in our confocal microscope images.

B. Recent Attempts at Network Segmentation

First we note that several commercially available software packages that perform some form of filament tracing in 2D or pseudo 2D cells, including [1], [14], [29] exist. We note, however, that the functionality of these softwares is limited to tracing structures similar to neurons where the resolution of the image is higher relative to the filaments (neurons) being traced. As a result the neuronal branches are clearly separated and the geometry of the branch cross sections are relatively uniform. In contrast, our images are populated with dense filament networks where the filament thickness is much below the optical resolution of fluorescent imaging. The result is optical blurring, low signal-to-noise ratio, and ambiguity in delineating junctions and intersections in a dense network. Most of the commercially available softwares use techniques such as finding unambiguous and clearly separated *seed points* along well separated filament branches, which are infeasible at the resolution of confocal images we work with.

Several researchers have pursued approaches for measuring partial topological parameters of filament networks from 2D images of cells. In [20] the authors use Fourier-based methods to estimate actin stress fiber orientation while the authors in [27] employ fractals to analyze cytoskeletal structure. In [11] the authors propose an improved orientation measurement compared to [20], [27] by accumulating image gradients into histograms defined over local image windows. Weichsel red *et al.* [31] proposes a similar method to Karlon *et al.* [11] where they calculate local coherency of the structure tensor in order to estimate the principal orientation of filaments. We note that although these estimated orientations have higher order information, calculations are independent of any actual segmentation of the actin fibers and are derived from image properties that relate to network topology only indirectly.

Lichtenstein *et al.* [13] develop a generative model for detection of filament pixels in fluorescence microsope images. This process is statistically amenable, but it does not explicitly address network geometry. Shariff *et al.* [26] also investigate a generative approach combined with indirect (inverse) estimation of the generative model to estimate basic parameters (number, mean length) from live and fixed cells. Fleischer [3] propose an interesting methodology for measuring actin network morphology by fitting geometric tessellation models to actin network images. Finally, Xu *et al.* [32] uses multiple open active contours to segment *in-vitro* actin filament populations. This method can provide individual filament information. However, contour merging and splitting rules (necessary operations in this method) are difficult to prescribe.

Multiple studies directly use 3D electron tomography images of actin and microtubule networks and use iterative point based 3D tracing algorithms to track filament centerlines ([21], [22], [30]). The most critical ingredient needed in this approach is a very high voxel resolution so that several 2D slices are available even in the flat regions close to the cell

cortex for the unamiguous tracking of individual filaments. However, this level of image oversampling may be infeasible due to photobleaching, photodamage or timescales of cellular events, as in live cell imaging. In presence of lower resolution imaging such as our confocal images, suboptimal algorithms that infer filament lengths from 2D projections of the cells are often the only choice. Moreover, in very flat cells as is the case with our datasets, the error in length computation due to projection along the microscope optical axes is negligible.

C. Our Algorithm and Contributions

Here we describe a method that can automatically localize and extract filaments from microscopy images. The method takes as input segmented cell images and outputs the centerline locations for the filaments, as well as an estimated list containing individual filaments (including junctions and intersections). This list of filaments can be subsequently used for various important computations such as filament curvature distributions, local connectivity and topology, orientation distributions, placement within the cell, number of filaments, as well as the filament length distribution in that cell. Our method builds on prior work consists of first estimating the locations of the centerlines of each filament using the reverse diffusion-based method described in [23]. Given a point cloud of centerlines, a new segment connectivity method based on integer programming is then used for extracting individual filaments.

We have applied our algorithm to estimate filament length distributions from simulated data, as well as real microscopy images of DNA filaments. We demonstrate both qualitatively and quantitatively the accuracy of our method in enumerating a complete list of filaments from a complicated filament network in an image. Fig. 1(A) demonstrates the conceptual building blocks of our algorithm, while Fig. 1(B) contains an example of the main blocks applied to an image of actin filaments [28]. Finally, as an example application, we use the method to extract filament distributions of actin filaments in *Arabidopsis* guard cells fluorescently labeled with 18 kinds of organelle markers in the Live Images of Plant Stomata (LIPS) database [7]. Specifically, we have used samples from the LIPS III database, which hosts a collection of maximum intensity projections of the fluorescent serial optical sections; we have chosen the actin microfilament channel for our experiments. In addition, we have also shown a practical use of our method in estimating change of DNA filament lengths as a function of exposure to UV radiation as shown in the first panel (Fig. 1) in the paper [16]. Application of our algorithm to the time series AFM panels of deposited DNA automatically measures and validates the hypothesis that the length of DNA filaments decrease with increasing exposure to UV radiation. A detailed explanation will follow in section III.

We note here that our filament extraction process currently is designed for enumerating single strands of elements. In many realistic scenarios, cytoskeletal filaments such as actin can form bifurcation points along their lengths. However, reliable estimation of branching points in a dense network is a challenging mathematical problem in itself and we reserve this useful addition to our algorithm for our next developmental cycle.

Section II describes our method. We start by providing a brief overview of the filament centerline extraction framework we use, followed by the integer programming-based

solution for individual filament extraction. Section III contains results that quantitatively and qualitatively evaluate the performance of our algorithm. We conclude the paper by summarizing and discussing the main findings.

II. Methods

As mentioned above, our method consists of two steps: filament centerline localization via reverse diffusion, and filament extraction via integer programming. We start by providing a brief overview of the centerline estimation approach below (more details are available in [23]) followed by a detailed description of the filament extraction method.

A. Filament centerline localization

The algorithm for filament localization utilizes a matched filter-based approach to first estimate the pixel positions that are likely to contain a filament and then proceeds by 'evolving' the detected points so as to converge them towards local optima (crests) in the image. As opposed to standard binary thinning approaches, the method utilizes the intensity values of the image to guide the filament thinning process.

Any filament recovery process has to first distinguish the fluorescence signal from acquisition noise, and also determine whether a fluorescence signal belongs to a filamentous protein or is simply background clutter from very short filaments. The method developed in [23] accomplishes this by finding regions that at the same time 1) have the 'appearance' of a filament type structure and 2) have relatively high intensity values (a fluorescent filament has higher intensity than a background pixel). A set of filament segment models (sampled at the same resolution of the given image) is used as a detection filter-bank. The filter-bank contains filament models of multiple lengths, orientations, and curvatures. The product of the raw image intensity and the normalized cross correlation of the image with each model filament gives the likelihood of the pixel belonging to a valid filament. The idea is to capitalize on the fact that filamentous regions usually have high intensity, as well as tend to 'look' like a linear structure. We note that our use of the word 'likelihood' here is a slight abuse of statistical nomenclature, given the fact that our likelihood values are not true probabilities (do not contain values only between 0 and 1). Nonetheless, the approach was found to be useful in discriminating between filamentous and background [23] regions.

A mathematical equation describing this process is given by

$$\mathscr{L}(\vec{x}) = I(\vec{x}) \max_{s,\theta,a} \{I \bigstar f_{s,\theta,a}(\vec{x})\} \quad (1)$$

where $I(\vec{x})$ is an input image, $\vec{x}$ a 2D image coordinate, and parameters s, θ, a corresponding to the scale, angle, and curvature or the models in the filter bank. Thus, the likelihood estimation procedure also outputs an angle θ corresponding to the local orientation of any filament present in pixel coordinate $\vec{x}$ (see [23] for more details). We denote this orientation vector, at location $\vec{x}$, by $O(\vec{x}) = [\sin \bar{\theta} \cos \bar{\theta}]^T$. The orientation vector field is then

'smoothed' (via local averaging process [23]) to minimize the effects of spurious detections. The smooth orientation field is denoted as $\hat{O}(\vec{x})$.

The next step in the process is to threshold $\mathcal{L}(\vec{x}) > a$ to obtain the list of pixels in image I that are most likely to contain a filament. The set of coordinates that satisfy this inequality are denoted as $\vec{X} = [x_1, x_2, \cdots, x_N]$ and serve as starting 'particles' to which we apply our reverse diffusion process. However, we constrain the movement of the particles to be along the direction perpendicular to $O(\vec{x})$, denoted by $\hat{O}(\vec{x})$. The differential equation describing this process is given by:

$$\frac{\overrightarrow{dx_i(t_i)}}{dt} = \hat{O}(\overrightarrow{x_i}(t_i)). \quad (2)$$

where time t_i here is an artificial parameter allowing us to specify the iterative maximization of the following objective function:

$$E(t_1, t_2, \ldots, t_n) = \lambda \sum_{i=1}^{n} \mathcal{L}(\overrightarrow{x_i}(t_i)) - (1-\lambda) \sum_{j=1}^{n} \sum_{k=1}^{n} \mathcal{K}(\overrightarrow{x_k}(t_k), \overrightarrow{x_j}(t_j))(\overrightarrow{x_k}(t_k) - \overrightarrow{x_j}(t_j))^2, \quad (3)$$

where $\lambda \in [0, 1]$ is a parameter that weights the two terms in the equation above, and $\mathcal{K}(x_i, x_j) = 1/\sqrt{2\pi\sigma^2} e^{-(x_i - x_j)^2/2\sigma^2}$.

Equation (3) is maximized using a standard gradient ascent method. It can be shown [23] that the iterative process can be written as

$$\overrightarrow{x_i}(t_i + \Delta t_i) = \overrightarrow{x_i}(t_i) + (\tau\lambda(\nabla \mathcal{L}(\overrightarrow{x_i}) \circ \hat{O}(\overrightarrow{x_i})) - \tau(1-\lambda)(\sum_{k=1}^{n} (4\mathcal{K}(\overrightarrow{x_i}, \overrightarrow{x_k})(\overrightarrow{x_i} - \overrightarrow{x_k}) \quad (4)$$
$$+ 2\frac{\partial \mathcal{K}(\overrightarrow{x_i}, \overrightarrow{x_k})}{\partial t_i}(\overrightarrow{x_i} - \overrightarrow{x_k})^2))\hat{O}(\overrightarrow{x_i}).$$

The second panel in Fig. 1 shows the output of the reverse diffusion step when applied to the actin stained image in panel one of the same image. As explained in [23], given a small enough step size τ, the algorithm is guaranteed convergence given that the cost function is bounded.

B. Extraction of Individual Filaments from the Centerline Pixel Cloud

We adopt a disambiguating technique that aims at finding a partitioning in the set of paths that does not violate certain constrains and parameters (described below) that are contrary to their natural arrangement in the biological environment. The approach first starts by generating a tree connecting all the centerline pixels into a centerline graph. Since only the

bifurcation nodes in the tree are the possible intersections of the filaments estimated by the reverse diffusion step in our algorithm, we are reasonably sure that adjacent non-bifurcating nodes in the tree belong to the same filament. Therefore, we remove all bifurcation nodes in the tree and start with the resultant non-intersecting set of connected chains as a first approximation to the filament set. Thereafter, we iteratively join adjacent filament fragments into longer filament segments such that an overall energy configuration of the filament set is minimized (described in more detail later) while simultaneously forbidding unlikely joining of segments (for example, joining two adjacent segments that are too far apart or that have end tangents highly mis-aligned).

The overall idea of iteratively combining shorter filament fragments into longer filaments is illustrated in Fig. 2. The steps are described in detail in the following sections.

1) Decomposition into Non-intersecting Filament Segments—We infer the local centerlines by connecting the converged centerline particles from the diffusion step through a minimum spanning tree (MST) [6], where each converged particle is treated as a graph node and the edge distance between any two nodes is the simple Euclidean distance separating them. Note, however, that the MST will also connect every particle with every other particle via some path. As a result, two completely separate filaments, although far apart in distance, might be connected by a spurious path. Consequently we assume that for a dense set of nodes such as the output from the reverse diffusion step, two nodes on one filament centerline and connected by the MST cannot be too far apart, therefore we disconnect edges in the MST that are longer than a prescribed length δ.

After this step, we have a set of smaller particle trees, each of which represent a set of intersecting filaments. Since only the bifurcation nodes in the tree(s) are the possible intersections of the filaments estimated by the previous reverse diffusion step, we are reasonably sure that adjacent non-bifurcating nodes in the tree belong to the same filament. Therefore, we delete *all* bifurcation nodes in the tree(s) and start with the resultant non-intersecting set of filament segments (comprised of single connected chains of particles, joined end-to-end) as a first approximation to the filament set. Many of these filament segments (connected chains) belong to the same filament and have been disconnected as a result of deleting the bifurcation nodes in the tree(s). A systematic procedure should now ensure that we combine correct non-intersecting segments to build a whole filament in the subsequent steps. The construction of the non-intersection filament segments is illustrated in Fig. 2(a)–(c).

Suppose the set of non-intersecting filament segments after disconnecting the bifurcation nodes in the MST be

$$\mathcal{S} = \{\mathbf{s}_i, i = 1, \cdots, n\} \quad (5)$$

where $\mathbf{s}_i$ is the ith segment given by a curve of m_i points in order, i.e.

$$\mathbf{s}_i = \{\vec{x}^i_1, \cdots, \vec{x}^i_{m_i}\}. \quad (6)$$

Here we have renumbered the converged centerline pixels from (4) as $\vec{x}^i_j$ so that we can identify the jth centerline pixel from the ith segment. As previously noted, we can reasonably assume that every final synthesized filament will be a combination of one or more of the segments in $\mathcal{S}$ (from (5)). The challenge is therefore to partition $\mathcal{S}$ into non-intersecting sets, and combine segments inside each partitioned set appropriately to synthesize the final filaments, all done in a manner that satisfies biological and geometric constraints.

2) Parameters for Combining Filament Segments—Before describing the criterion for optimal global synthesis of a set of longer filament segments from a set of shorter filament segments, let us explain in detail how we construct one longer segment by combing a few adjacent shorter segments. Consider three segments, $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$, that we might combine to make a single segment. Since each segment $\mathbf{s}_i$ consists of pixels indexed in order (from one end of the segment to the other), we might name the two endpoints of the segment $\mathbf{s}_i$ as $i(1)$ and $i(2)$ (the exact number does not matter). A combination of the three segments $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$ simply means traversing the pixels of the three segments in the order i, j and k, with a particular choice of endpoint to endpoint connection between the three segments. For example, $[i(1), i(2) - j(1), j(2) - k(2), k(1)]$ may be one choice of endpoint connections in which one can traverse the three segments in order.

It is easy to check that keeping the orientation of the middle segment j fixed, there are only four possible ways ([1,2-1,2-2,1], [1,2-1,2-1,2], [2,1-1,2-2,1] and [2,1-1,2-1,2]) in which endpoint connections between $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$ can be established for traversing the three segments in order (flipping the orientation of j simply results in the mirror images of the already obtained traversal orders). By an entirely similar argument, two segments $\mathbf{s}_i$ and $\mathbf{s}_j$ can be joined with two choices of endpoint connections.

Now, for each choice of endpoint connection between three segments in order (or two segments in order), we can define a traversal cost c of the three (two) segments; for example, in this work, for an ordered set of pixels $\{\vec{y}_1, \cdots, \vec{y}_m\}$ (representing a segment) obtained by a particular ordered traversal of the three (two) segments, we have fit a cubic smoothing spline to the set with spline nodes placed at unit resolution and then calculated the joining cost of the pixel set by adding up curvature energies at the individual spline nodes (the *curvature energy* can be calculated by summing up the squared second derivatives of the spline coordinates with respect to the spline arc-length at the spline nodes). A specific choice of spline function and the form of the joining cost may depend upon the application at hand and the specific implementation, but the framework for defining a joining cost for segments is general.

Of course, as already mentioned, for three (two) segments, there are four (two) choices of endpoint connections. Therefore, given three (two) segments $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$ ($\mathbf{s}_i$ and $\mathbf{s}_j$), we define the *joining cost* $\mathbf{c}(\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\})$ of the segments as the minimum traversal cost c of three (two) segments with all possible endpoint combinations. Although the minimum endpoint configuration might not be unique, it is clear that the joining cost $\mathbf{c}$ is.

3) Valid Combinations of Filament Segments—With the above-mentioned parameters for combining segments, let us investigate the possible ways of choosing segments that are appropriate for combination into a longer segment. Let $dist(\mathbf{s}_i, \mathbf{s}_j)$ denote the minimum distance between any endpoint of $\mathbf{s}_i$ and any endpoint of $\mathbf{s}_j$. For a segment $\mathbf{s}_i \in \mathcal{S}$, define the neighborhood set $\mathcal{N}(\mathbf{s}_i)$ of $\mathbf{s}_i$ as the set of segments $\mathcal{N}(\mathbf{s}_i) = \{\mathbf{s}_j \in \mathcal{S} \mid dist(\mathbf{s}_i, \mathbf{s}_j) \leq \delta\}$, where δ is input by the user and controls the extent of gaps between neighboring segments that the algorithm can bridge.

For a set of segments $\mathcal{S}$, the valid segment combinations are represented as sets $\mathcal{V}_l(\mathbf{s}_j) \subseteq \mathcal{S}$, where each $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k$ for some $\mathbf{s}_i, \mathbf{s}_k \in \mathcal{N}(\mathbf{s}_j)\}$, or $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_i, \mathbf{s}_j$ for some $\mathbf{s}_i \in \mathcal{N}(\mathbf{s}_j)\}$, or $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_j\}$. Note that single segments $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_j\}$ are prescribed as valid combinations because a single segment might be incompatible with any other segment in the set. We will elucidate this point later in more detail. The total number of valid sets $\mathcal{V}_l(\mathbf{s}_j)$ (i.e., the maximum value of the set index l) will depend on the proximity of segment $\mathbf{s}_j$ with other segments in $\mathcal{S}$.

Along with the valid segment combinations, we also prescribe the *combination cost* $\mathcal{C}(\mathcal{V}_l(\mathbf{s}_j))$ as follows:

1. $\mathcal{C}(\mathcal{V}_l(\mathbf{s}_j)) = \mathbf{c}(\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}) - 2b$ when $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k | \mathbf{s}_i, \mathbf{s}_k \in \mathcal{N}(\mathbf{s}_j)\}$,
2. $\mathcal{C}(\mathcal{V}_l(\mathbf{s}_j)) = \mathbf{c}(\{\mathbf{s}_i, \mathbf{s}_j\}) - b$ when $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_i, \mathbf{s}_j | \mathbf{s}_i \in \mathcal{N}(\mathbf{s}_j)\}$,
3. $\mathcal{C}(\mathcal{V}_l(\mathbf{s}_j)) = \mathbf{c}(\{\mathbf{s}_j\})$ when $\mathcal{V}_l(\mathbf{s}_j) = \{\mathbf{s}_j\}$. The cost $\mathbf{c}(\{\mathbf{s}_j\})$ is simply taken as the traversal cost c of the set of pixels in the set $\mathbf{s}_j$ itself.

The term b in the definition for $\mathcal{C}(\mathcal{V}(\mathbf{s}_j))$ is called the *bond energy*. Since joining three (or two) segments will definitely increase the additive curvature energy (given by $\mathbf{c}$) of the combined segment, the bond energy term determines the chance of a particular combination to decrease the overall curvature energy of the system. The bond energy b therefore is a positive scalar pseudo-energy parameter that dictates whether the combination of three (two) segments $\mathbf{s}_i$, $\mathbf{s}_j$ and $\mathbf{s}_k$ ($\mathbf{s}_i$ and $\mathbf{s}_j$) can reduce the total additive curvature energy of the new combined system as opposed to leaving the original segments uncombined with their individual curvature energies $\mathbf{c}(\{\mathbf{s}_j\})$. The bond energy is prescribed per 'bond' or endpoint connection between two segments, hence for combining three segments we subtract $2b$ from the combined curvature energy. We will give a numerical method to calculate the bond energy at the end of this section. A schematic representing the various possible ways of legal combination of a set of adjacent shorter filaments into a set of longer filaments is shown in Fig. 2(d).

4) Iterative Combination of Filaments—In order to synthesize a complete list of filaments from the short filament segments in $\mathcal{S}$, we adopt a maximal set partitioning

procedure using a variant of binary integer programming [18]. We iteratively combine the segments inside $\mathscr{S}$, calling the updated sets $\mathscr{S}^1, \mathscr{S}^2, \cdots, \mathscr{S}^t, \mathscr{S}^{t+1}, \cdots, \mathscr{S}^T$ where t denotes the iteration index with $t = T$ at convergence.

We now have all the ingredients to iteratively update a set of segments $\mathscr{S}^{t+1}$ from $\mathscr{S}^t$. At every iteration, $\mathscr{S}^t = \{\mathbf{s}_i^t, i = 1, \cdots, n_t\}$, we build the collection of all possible valid combination sets $\mathscr{M}_t = \{\mathscr{V}_l(\mathbf{s}_j^t)\}\ \forall \mathbf{s}_j^t \in \mathscr{S}^t$. We take care to avoid duplicate sets, i.e., whenever $\mathscr{V}_{l_1}(\mathbf{s}_j^t) = \mathscr{V}_{l_2}(\mathbf{s}_i^t)$ for some l_1, l_2, $\mathbf{s}_i^t$ and $\mathbf{s}_j^t$, we omit either $\mathscr{V}_{l_1}(\mathbf{s}_i^t)$ or $\mathscr{V}_{l_1}(\mathbf{s}_j^t)$ from $\mathscr{M}_t$. Suppose we list all the combinations in $\mathscr{M}_t$ and assign each combination a binary variable β_m, $m = 1, \cdots, |\mathscr{M}_t|$. This simply means that every $\mathscr{V}_l(\mathbf{s}_j^t) \in \mathscr{M}_t$ has a binary variable $\beta_m(\mathscr{V}_l(\mathbf{s}_j^t))$ attached to it. Similarly, given the binary variable β_m, let us denote the combination set attached to it as $\mathscr{V}_{\beta_m}$. Also, let us build the *indicator matrix* $A^t \in \mathbb{R}^{|\mathscr{S}^t| \times |\mathscr{M}_t|}$ as follows - if $\mathscr{V}_{\beta_m} = \mathscr{V}_l(\mathbf{s}_j^t) = \{\mathbf{s}_i^t, \mathbf{s}_j^t, \mathbf{s}_k^t\} \in \mathscr{M}_t$, then the ith, jth and kth rows of the mth column of A^t are 1, all other rows of the mth column of A^t are zero. In other words, $A^t(i, j) = 1$ iff the jth combination set includes segment i in it, otherwise $A^t(i, j)$ is 0. For a combination set $\mathscr{V}_{\beta_m}$ that has two (or one) constituent segments, we put the corresponding rows of the mth column as one, with all other rows being zero. A^t thus codifies all the combination sets in $\mathscr{M}_t$ such that each column of A^t represents a distinct combination set with the chosen segments being denoted with a one in the corresponding rows. For example, if $A^t(3, 4) = 1$, then the 4th combination set includes segment 3 in it.

Our goal is to choose a subset of combinations from $\mathscr{M}_t$ that minimizes some global combination cost, at the same time selecting every segment $\mathbf{s}_j^t \in \mathscr{S}^t$ once and only once across all the chosen combination sets. This is expressed mathematically as the solution of the following binary integer programming problem:

$$\min_{\beta_m = \{0,1\}} \sum_{m=1}^{|\mathscr{M}_t|} \beta_m \mathscr{C}(\mathscr{V}_{\beta_m}) \quad (7)$$

subject to

$$\sum_{m=1}^{|\mathscr{M}_t|} A^t{}_{im} \beta_m = 1, (i = 1, \cdots, n_t)$$

Eq. (7) simply means we choose a subset of $\mathscr{M}_t$ given by those elements of $\mathscr{M}_t$ whose corresponding indicator variables $\beta_m = 1$. After the solution to (7) has been obtained, suppose for some m, $\beta_m = 1$, and $\mathscr{V}_{\beta_m} = \mathscr{V}_l(\mathbf{s}_j^t) = \{\mathbf{s}_i^t, \mathbf{s}_j^t, \mathbf{s}_k^t\}$, then a new segment $\mathbf{s}_p^{t+1}$ in the next iteration step is obtained by actually combining $\mathbf{s}_i^t$, $\mathbf{s}_j^t$ and $\mathbf{s}_k^t$ following the endpoint connections that generated the joining cost $\mathbf{c}(\{\mathbf{s}_i^t, \mathbf{s}_j^t, \mathbf{s}_k^t\})$. A similar argument holds for a $\beta_m = 1$ whose combination set consists of two or one segment. More specifically, in case of a single segment in the combination set, we retain the original segment as is, and nothing is combined with it at all.

This process is continued until convergence, i.e., until $\mathcal{S}^t = \mathcal{S}^{t-1}$. The set of combined segments $\mathcal{S}^t = \{\mathbf{s}_p^t, p = 1, \cdots, n_t\}$ is output as the final list of extracted filaments. It is worth noting that at every step we limit our valid combinations of segments to end-to-end connection between three segments at the most. This is because at every iteration step, we allow the segments to grow one step at both ends, if possible. The best combination of growing is selected by the optimization step, and subsequent growth at both ends is decided by another optimization at a later iteration. It is easy to see that at the final iteration step, only those combination sets will be selected which have single elements in them. In other words, all the segments from the previous iteration step are retained since no further combinations are possible. The third panel in Fig. 1 shows the output of the filament extraction step when applied to the centerline localized image in panel two of the same image. An intermediate iteration step that chooses the optimum joining of a set of adjacent shorter filaments into a set of longer filaments from a range of choices in shown in Fig. 2(d)–(e). The complete idea of iterative combination of adjacent filament segments is presented in a flowchart in Fig. 2(f).

5) Calculation and Significance of the Bond Energy *b*—The bond energy b used to derive the combination cost $\mathcal{C}(\mathcal{V}_j(\mathbf{s}_j))$ in section II-B3 conceptually encapsulates the familiar parameters of linking distance and linking angle in some edge-linking algorithms, albeit in an alternative energy formulation. We would ideally want to join two previously unconnected filament segments into a bigger filament filament if and only if they lie *close enough* and visually perceived as as natural continuation of each other (or in other words, they are *aligned similarly*). Amongst several choices of segment pairs that can be possibly combined, we also need to find those combination pairs that lead to optimal combination, as in, they reduce some global energy configuration as in (7). The bond energy translates these geometric considerations into an additive energy constraint that fits well into an energy minimization approach of combining two segments (7). It simply decides whether the total curvature energy of two or three segments, when they are joined, is reduced enough compared to sum of the individual curvature energy of the segments. If not, the optimization step in (7) will likely not choose such a combination and will keep the individual segments unjoined.

Thus, an accurate estimate of b is vital to the success of our method. A fairly reasonable estimate of the bond energy can be made based on two physical parameters that need to be input by the user. The concept is illustrated in Fig. 3. Suppose that it is unlikely that segments that are separated by more than an endpoint to endpoint distance of δ come from the same filament. Also, if 1 and 2 in Fig. 3 represent the endpoints of two different filaments separated by a distance δ, let θ_1 and θ_2 represent the tangent directions at the endpoints of segment 1 and 2 respectively. Note that θ_1 is measured counterclockwise about point 1 and θ_2 is measured clockwise about point 2; this is to ensure that the tangents are on the same side of the line joining 1 and 2. In case the tangents are both on the opposite sides of the line joining 1 and 2, θ_1 is measured clockwise about point 1 and θ_2 is measured counterclockwise about point 2. We consider the joining of two endpoints whose tangents are on opposite sides of the line joining two endpoints highly unlikely.

Once the the relative position of the two endpoints 1 and 2 are fixed, we can calculate the cost of joining a cubic spline $E(\delta, \theta_1, \theta_2)$ between two points 1 and 2, having a relative separation of δ and relative endpoint tangents $\tan(\theta_1)$ and $\tan(\theta_2)$ with the configuration as in Fig. 3, where the cost is calculated in the same way as the cost of joining segments **c** is calculated in section II-B3. Then the bond energy b is calculated as

$$\underset{\delta \in [0, \delta]\theta_1, \theta_2 \in [0, \theta]}{\operatorname{argmax}} E(\delta, \theta_1, \theta_2). \quad (8)$$

Eq. (8) means that bond energy b is simply the maximum cost that we might incur while joining two endpoints whose relative tangents are at most misaligned by an angle θ ($\theta_1, \theta_2 \in [0, \theta]$ in (8)) and which are separated by at most a distance δ. Therefore, our algorithm needs two inputs from the user, the maximum segment linking distance δ and a maximum mismatch in orientation θ - both can be expected to be known in advance reasonably accurately from the biological constraints of the problem. In order for two endpoints from two different segments to be combined, the additional cost incurred by the combination process in section II-B3 must be strictly less than b; any misalignment or separation that causes the extra combination energy to be more than b cannot be balanced by a subtraction of b in the formula for $\mathscr{C}(\mathscr{V}(\mathbf{s}_j))$ in section II-B3.

III. Experiments

Our filament localization algorithm has been tested on a database of real and simulated images to test for both accuracy and applicability. To generate our filter bank, we used a filament width $w = 2$ pixels for the simulated image database and $w = 0.4\mu m$ for the real image database, and a fine sampling of curvatures, orientations and scales in our filter bank. In all results shown, the threshold parameter for the likelihood function was set to $\alpha = 0.1$. The width of the Gaussian function $\mathscr{K}(\vec{x}_i, \vec{x}_j)$ was set to $\sigma = 3$ pixels for the simulated images and $\sigma = 0.6\mu m$ for the real image database.

A. Validation on simulated images

We have tested our filament extraction methodology on 100 simulated images of size 128 × 128 pixels consisting of filament networks that vary in complexity and filament density. The real cytoskeletal filament generation process is complex and not fully understood to the level of accurate simulations. Therefore, a simple assumption of white filament centerlines on a black background served as the ideal image, and it was subsequently convolved with a psf that approximated a real microscope.

The length of individual filaments are drawn randomly from a Gaussian distribution of mean 50 pixel lengths and a standard deviation of 3 pixel lengths. The starting point of each filament in each image is chosen randomly, and the filament is allowed to grow to the sampled length incrementally, with the direction of each incremental addition (the direction is measured as an angle to the previous growth direction) selected uniformly randomly between zero and 20°.

The maximum filament count is 37 per image and four images per filament count are generated. The maximum allowable separation δ between filament endpoints have been set to 2 pixels and the maximum tangent mismatch θ has been set to 20° since we know the parameters beforehand from the generation process. Panels (a), (b), (c) and (d) of Figure 4 show four simulated images.

Fig. 5 shows the result of our filament extraction methodology on the simulated image database. We calculate the relative error for *filament position* in every image (shown in blue in Fig. 5(i)) as the sum of two distances - (A) the distance between every pixel of every simulated filament and the nearest pixel from the best extracted filament that matches the simulated filament, relative to the total length of the simulated filaments in the image and (B) the distance between every pixel of every extracted filament and the nearest pixel from the best simulated filament that matches the extracted filament, relative to the total length of the simulated filaments in the image. The relative error for *filament mean length* in every image (shown in red in Fig. 5(i)) is calculated as the absolute difference of the mean filament length of the simulated image and the mean filament length from the corresponding extracted filament image, relative to the total length of the simulated filaments in the image. The relative error in *filament count accuracy* in Fig. 5(j) is simply the error in estimation of the total filament count divided by the total filament count.

Figs. 5(a)–(d) show examples of the simulated filament images. Panels (e), (f), (g) and (h) shows the corresponding filament extracted images. Different colors in (e)–(h) denote different filaments, although colors are not unique. Fig. 5(i) shows the filament extraction accuracy in percentage error and Fig. 5(j) shows the relative error in determining the total filament count in an image. It can be noticed that apart from regions in Figs. 5(a)–(d) where the filament distribution is too dense for correctly identifying separate filaments for even the human eye, our extraction method does a reasonably accurate job of extracting the individual filaments by disambiguating filament intersections. The filament extraction accuracy, as described above, remains within a acceptable value of 2% (for positional error) and 1.7% (for mean length error) in Figs. 5(i) despite variation in filament density (number of filament per image). Additionally, apart from one image where the relative error in filament count is high, the algorithm performs well in estimating the total number of filaments from the error histogram in Fig. 5(j).

B. Real Filament Images

We have applied the filament localization step to image datasets of cells. Our real cell image dataset consists of HeLa cells (Fig. 4(e)) that have been fixed and labeled with rhodamine phallodine that preferentially binds to actin filaments. We have used another publicly available database of *Arabidopsis* guard cells fluorescently labeled for actin microfilaments [7]. Figs. 4(f) and 4(g) show two maximum intensity projections of plant stomata guard cells from the LIPS-III subdatabase in [7]. The value of the maximum allowable gap between filament endpoints, or δ has been set equal to the standard deviation σ of the Gaussian neighborhood kernel from (3), which is fixed at $\sigma = 0.6\mu m$. The maximum allowable tangent mismatch θ has been set to 20°, which is a reasonable tangent mismatch of separate actin filaments in our problem setup.

Figure 6 shows the result of applying our filament localization and extraction method on the raw images. The first column shows the three raw images of HeLa cell (first row) and plant stomata guard cells (second and third row). The second column shows the localized centerlines of the actin filaments after the reverse diffusion step described earlier of the corresponding images (red pixels overlaid on the raw images). The third column shows the corresponding filament extracted images, where individual filaments are shown in different colors. Note that the colors are not unique to each filament. From the results, it is clear that our filament localization and extraction method does a reasonably accurate job of predicting individual filament from the images.

The choice of the various filament localization and extraction parameters are somewhat crucial to the performance of our algorithm. The filament localization step requires a choice of neighborhood size and a clustering threshold, but the mechanics of reverse diffusion make the localization process relatively robust to slight mis-calibrations of the parameters; the corresponding experimental verification has already been provided in [23].

In this work, we have experimented with slight mis-calibrations of the maximum segment gap δ and maximum orientation mismatch θ for calculating the bond energy. A visual inspection of Fig. 7 shows that our algorithm is relatively robust to mis-calibrations and imperfect assumptions of the filament linking parameters.

It is important to note that a considerable mis-estimation in the afore-mentioned filament linking parameters might produce inaccurate results; but as is the norm with almost all biological problems, a reasonable value for the algorithmic parameters must be decided upon *apriori* using cues from the real biological problem. In our case, it is not a difficult task to come up with reasonable values of the maximum edge gap and orientation parameters, and it is our belief that slight variation of the actual algorithmic parameters around the assumed values will produce similar (if not exactly same) results.

As mentioned earlier in section I-B, intensity based centerline tracing algorithms are a popular choice in very high resolution 3D tomography images of cytoskeletal structures ([21], [22], [30]). Here, the voxel size ($0.3nm$) is much smaller compared to the actin filament diameter ($7nm$). Consequently, a template based filament detection step similar to the enhancement step in our algorithm is used to enhance the filamentous structures in the image volume. This step is followed by a seed following algorithm that depends on the likelihood of adjacent voxels belonging to a continuous filament satisfying certain geometric smoothness criteria (see [21]). This approach is not suitable in the case of our microscopy images at much lower resolutions that show significant optical blurring, filament bundling and filament intersections (due to the pseudo-2D nature of the actin dynamics on very *flat* regions of the cells).

To demonstrate the inapplicability of simple filament tracing algorithms such as [21] in our case, we have reimplemented [21] for a 2D filament tracing scenario with the results shown in Fig. 8. The algorithm in [21] is crucially dependent on many smoothness parameters and intensity thresholds, and even with a suitable choice of all parameters, the maximum number of seed initializations based on the sorted maximum correlations of the pixels can drastically

affect the extracted filaments. For example, the maximum number of seed initializations are set to 100 and 200 in Figs. 8(c) and 8(d), and the algorithm finds 44 and 83 filaments of length more than 2 voxels respectively. The non-uniformity of our low resolution images and filament intersections cause [21] to mis-trace the centerlines within a bundle and confound the tracing at intersections. Compared to [21], our algorithm gives a much more reasonable filament extraction in Fig. 8(b).

In addition, in Fig. 9, we have applied our algorithm for an automated estimation of change of DNA length with exposure to UV radiation as described in [16]. Fig. 1 in [16] investigated single DNA molecules on mica surfaces and the influence of UV irradiation on the geometric structure of the attached DNA. A pertinent measurement of geometric structure being the length of the DNA contours, we applied our algorithm to automatically output the mean length of the DNA strands in Figs. 9(A)–(C). For verification, we have also manually segmented the DNA panels and computed te centerline lengths of the DNA strands. Fig. 9 shows the manual estimation of the DNA strands (in red) vis-a-vis that computed by our algorithm (in blue). It can be seen that our algorithm correctly captures the general trend of decreasing DNA contour length with increase in UV irradiation (as verified in [16]). The normalization of mean length has been performed with respect to the image dimensions. This shows a practical application of our method to relevant problems of filament network extraction from other domains.

IV. Summary and discussion

We have described an approach for localizing and extracting cellular filaments from microscopy images. The approach builds on earlier work [23] and uses a reverse diffusion-based approach for localizing centerlines of filaments in a given image. The point cloud is then analyzed in an attempt to delineate individual filaments. The tree of centerline pixels output by our method can then be subsequently used in calculation of many important biological properties of filament networks in cells, such as filament length histograms, curvature distributions and orientation histograms.

We characterized the error of our algorithm in carefully constructed simulations. In these, it was shown that the error (both in terms of filament localization and filament counts) were on the order of a few percent. We demonstrate the feasibility of our approach by applying our tool to estimate actin filament networks in confocal microscope images of filament distributions in several cells. Extraction of filaments is possible even in the case of dense networks with complicated intersections and bifurcations. Experiments on a limited set of HeLa cells showed that carbon nanotubes may have an adverse effect in the configuration of actin networks in cells, with cells treated with a SWCNT solution showed on average shorter filaments. More data will be used in the future to confirm whether these results are statistically significant.

It is also important to highlight a few limitations of the algorithm. First we note that the implementation we use is appropriate for two dimensional microscopy images, while a cell is a 3D structure. We note that nothing in our methodology impedes the approach from being implemented in 3D as well. We note however, that most experimental data available for

quantifying filamentous structures are available in 2D images only [3], [11], [12], [19], [20], [26], [27], [31], [32]. We also mention that the computational complexity of the method is high; for 256 × 256 images of the actin filaments, the average runtime is 323 secs. This is primarily due to two reasons - (1) the reverse diffusion step for localizing centerlines evolves a differential equation on all candidate centerline particles and (2) the complexity of the binary integer programming in the iterative filament combination step grows multiplicatively with the number of non-intersecting filament fragments available as the first approximation to the final filament set. However, our algorithm was implemented in the Matlab [17] programming language, with extensive use of 'FOR' loops, which are notoriously slow. We are confident the time of computation, however could be significantly improved by implementing the code in a compiled (as opposed to interpreted) language.

Finally, we also clarify that the technique described here is most applicable when the complexity of the filament network and image resolution are such that individual filaments are visibly discerned. In some instances, such as when imaging microtubule structures with confocal techniques over the entire cell [26], this approach is not expected to produce useful results.

Acknowledgments

SB and GKR acknowledge support from grant NIH GM090033. The authors would like to acknowledge Dr. Kris Dahl from Carnegie Mellon University for input in useful discussions.

References

1. Bitplane Scientific Software., 2012.
2. Disanza A, Steffen A, Hertzog M, Frittoli E, Rottner K, Scita G. Actin polymerization machinery: the finish line of signaling networks, the starting point of cellular movement. Cell Mol Life Sci. 2005; 62(3):955–970. [PubMed: 15868099]
3. Fleischer, Frank, Ananthakrishnan, Revathi, Eckel, Stefanie, Schmidt, Hendrik, Ks, Josef, Svitkina, Tatyana, Schmidt, Volker, Beil, Michael. Actin network architecture and elasticity in lamellipodia of melanoma cells. New Journal of Physics. 2007; 9(11):420.
4. Giannone, Grégory, Dubin-Thaler, Benjamin J., Döbereiner, Hans-Günther, Kieffer, Nelly, Bresnick, Anne R., Sheetz, Michael P. Periodic lamellipodial contractions correlate with rearward actin waves. Cell. 2004; 116(3):431–443. [PubMed: 15016377]
5. Gonzales, R., Woods, R. Digital Image Processing. Addison-Wesley Publishing Company; 1992.
6. Graham RL, Hell Pavol. On the history of the minimum spanning tree problem. IEEE Ann Hist Comput. Jan; 1985 7(1):43–57.
7. Higaki, Takumi, Kutsuna, Natsumaro, Hasezawa, Seiichiro. Lips database with lipservice: a microscopic image database of intracellular structures in arabidopsis guard cells. BMC Plant Biology. 2013; 13(1):81. [PubMed: 23679342]
8. Holt, Brian D., Short, Philip A., Rape, Andrew D., Wang, Yu-li, Islam, Mohammad F., Dahl, Kris Noel. Carbon nanotubes reorganize actin structures in cells and ex vivo. ACS Nano. 2010; 4(8): 4872–4878. [PubMed: 20669976]
9. Huang, Bo, Babcock, Hazen, Zhuang, Xiaowei. Breaking the diffraction barrier: super-resolution imaging of cells. Cell. 2010; 143(7):1047–1058. [PubMed: 21168201]
10. Kanchanawong, Pakorn, Greenan, Garrett, Pasapera, Ana M., Ramko, Ericka B., Davidson, Michael W., Hess, Harald F., Waterman, Clare M. Nanoscale architecture of integrin-based cell adhesions. Nature. 2010; 468(7323):580–584. [PubMed: 21107430]

11. Karlon, William J., Hsu, Pin-Pin, Li, Song, Chien, Shu, McCulloch, Andrew D., Omens, Jeffrey H. Measurement of orientation and distribution of cellular alignment and cytoskeletal organization. Annals of Biomedical Engineering. 1999; 27:712–720. DOI: 10.1114/1.226 [PubMed: 10625144]
12. Li, Jieyue, Shariff, Aabid, Wiking, Mikaela, Lundberg, Emma, Rohde, Gustavo K., Murphy, Robert F. Estimating microtubule distributions from 2d immunofluorescence microscopy images reveals differences among human cultured cell lines. PLoS ONE. 2012; 7(11):e5029211.
13. Lichtenstein, Nurit, Geiger, Benjamin, Kam, Zvi. Quantitative analysis of cytoskeletal organization by digital fluorescent microscopy. Cytometry A. 2003; 54(1):8–18. [PubMed: 12820116]
14. Longair, M. Simple Neurite Tracer. 2010.
15. Loss LA, Bebis G, Parvin B. Iterative tensor voting for perceptual grouping of ill-defined curvilinear structures. Medical Imaging, IEEE Transactions on. Aug; 2011 30(8):1503–1513.
16. Lysetska M, Knoll A, Boehringer D, Hey T, Krauss G, Krausch G. Uv light-damaged dna and its interaction with human replication protein a: an atomic force microscopy study. Nucleic Acids Research. 2002; 30(12):2686–2691. [PubMed: 12060686]
17. Mathworks. MATLAB. 2012.
18. Nemhauser, GL., Wolsey, LA. Integer and Combinatorial Optimization. John Wiley & Sons; New York: 1988.
19. Ott A, Magnasco M, Simon A, Libchaber A. Measurement of the persistence length of polymerized actin using fluorescence microscopy. Phys Rev E. Sep.1993 48:R1642–R1645.
20. Petroll WM, Cavanagh HD, Barry PP, Andrews, Jester JV. Quantitative analysis of cell fiber orientation during corneal wound contraction. J Cell Sci. 1993; 104(2):353–363. [PubMed: 8505365]
21. Rigort, Alexander, Günther, David, Hegerl, Reiner, Baum, Daniel, Weber, Britta, Prohaska, Steffen, Medalia, Ohad, Baumeister, Wolfgang, Hege, Hans-Christian. Automated segmentation of electron tomograms for a quantitative description of actin filament networks. Journal of structural biology. 2012; 177(1):135–144. [PubMed: 21907807]
22. Rusu, Mirabela, Starosolski, Zbigniew, Wahle, Manuel, Rigort, Alexander, Wriggers, Willy. Automated tracing of filaments in 3d electron tomography reconstructions using¡ i¿ sculptor¡/ i¿ and¡ i¿ situs¡/i¿. Journal of Structural Biology. 2012
23. Basu S, Dahl KN, Rohde GK. Localizing and extracting filament distributions from microscopy images. Journal of Microscopy. 2013; 30(1):57–67.
24. Schwarz, Ulrich S., Gardel, Margaret L. United we stand–integrating the actin cytoskeleton and cell–matrix adhesions in cellular mechanotransduction. Journal of Cell Science. 2012; 125(13): 3051–3060. [PubMed: 22797913]
25. Selvin, Paul R., Syed, Raheem, Sobh, Nahil. Illinois tool: Fiona (fluorescence imaging with one nanometer accuracy). Dec.2010
26. Shariff, Aabid, Murphy, Robert F., Rohde, Gustavo K. A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images. Cytometry Part A. 2010; 77A(5):457–466.
27. Thomason, Donald B., Anderson, Otis, Menon, Vandana. Fractal analysis of cytoskeleton rearrangement in cardiac muscle during headdown tilt. Journal of Applied Physiology. 1996; 81(4):1522–1527. [PubMed: 8904563]
28. Wang, Yu-Li. Noise-induced systematic errors in ratio imaging: serious artefacts and correction with multi-resolution denoising. Journal of Microscopy. 2007; 228(2):123–131. [PubMed: 17970912]
29. Wearne SL, Rodriguez A, Ehlenberger DB, Rocher AB, Henderson SC, Hof PR. New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. Neuroscience. 2005; 136(3):661–680. [PubMed: 16344143]
30. Weber, Britta, Shtengel, Gleb, Prohaska, Steffen, Baum, Daniel, Hege, Hans-Christian, Müller-Reichert, Thomas, Hyman, Anthony A., Verbavatz, Jean-Marc. Automated tracing of microtubules in electron tomograms of plastic embedded samples of caenorhabditis elegans embryos. Journal of structural biology. 2012; 178(2):129–138. [PubMed: 22182731]

31. Weichsel, Julian, Herold, Nikolas, Lehmann, Maik J., Krusslich, Hans-Georg, Schwarz, Ulrich S. A quantitative measure for alterations in the actin cytoskeleton investigated with automated high-throughput microscopy. Cytometry Part A. 2010; 77A(1):52–63.

32. Xu, Ting, Li, Hongsheng, Shen, Tian, Ojkic, N., Vavylonis, D., Huang, Xiaolei. Extraction and analysis of actin networks based on open active contour models. Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on; 30 2011–april 2 2011; p. 1334-1340.

Biographies

Saurav Basu Dr. Basu is currently working as a Research Staff Member in the IBM India Research Laboratory at New Delhi, India. He develops algorithms in tomographic imaging, statistical signal processing and computational partial differential equations. Till July 2013, he was engaged in developing bioimage informatics retrieval systems in the Center for Bioimage Informatics, Carnegie Mellon University, USA. He was conferred a PhD in Image Analysis by the University of Virginia, USA, in 2011, where he worked with Prof. Scott T. Acton to develop mathematical algorithms for image recognition that leveraged techniques in differential geometry. He has been actively involved in the image analysis community as a researcher and reviewer for distinguished journals and conferences. His interests include application of differential geometry, machine learning, signal processing, optimization and PDEs to problems in imaging, computer vision, inverse problems and control.

Chi Liu Chi Liu is now a first year Ph.D. student in the Center for Bioimage Informatics (CBI), Carnegie Mellon University, under the supervision of Dr. G. K. Rohde. His research interests include image processing and analysis, machine learning and biomedical imaging.

Gustavo Kunde Rohde Dr. Rohde earned B.S. degrees in physics and mathematics in 1999, and the M.S. degree in electrical engineering in 2001 from Vanderbilt University. He received a doctorate in applied mathematics and scientific computation in 2005 from the University of Maryland. From 2005 to 2006 he worked as a post doctoral associate at the Naval Research Laboratory, Washington DC, where his work was funded by the National Research Council, National Academies. From 2007–2012 he was an assistant professor of biomedical engineering and electrical and computer engineering, at Carnegie Mellon University. In 2012 he became an associate professor at Carnegie Mellon University where he is a core faculty member of the Center for Biomage Informatics, and the Lane Center for Computational Biology. He is a member of the IEEE and ISAC, as well as an associate editor for the IEEE signal processing letters, and the BMC bioinformatics journals. He has published over 40 peer reviewed manuscripts in topics related to image processing and analysis, signal processing, pattern recognition, computational biology, biomedical imaging, and image-based pathology.

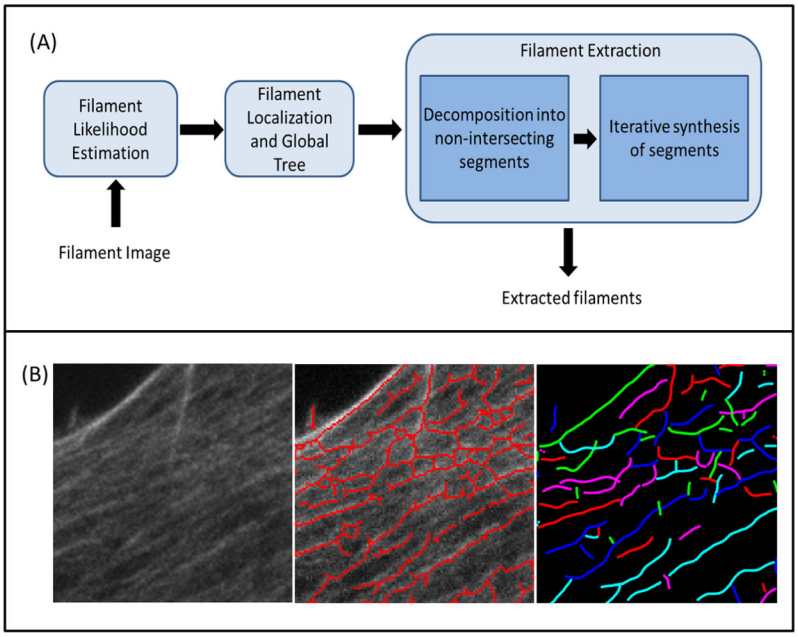

Fig. 1.

Overview of our filament extraction procedure. (A) A flow diagram showing the several sequential steps in our filament extraction algorithm from filament network images. (B) A real example demonstrating the extraction pipeline. The input image (leftmost panel) is first analyzed to determine the likelihood that a filament is present at any given pixel. The likelihood function is then thresholded to obtain an initial estimate of the filament locations. The initial estimate is 'evolved' with the constrained reverse diffusion-based algorithm described in the text so as to estimate the centerlines of each likely filament (second panel). The estimated filament centerlines are then decomposed into a set of non-intersecting short filament segments and an iterative filament combination process joins the short filaments into complete longer filaments disambiguating filament intersections; the individual filaments are shown in different colors (not unique) in the right most panel.

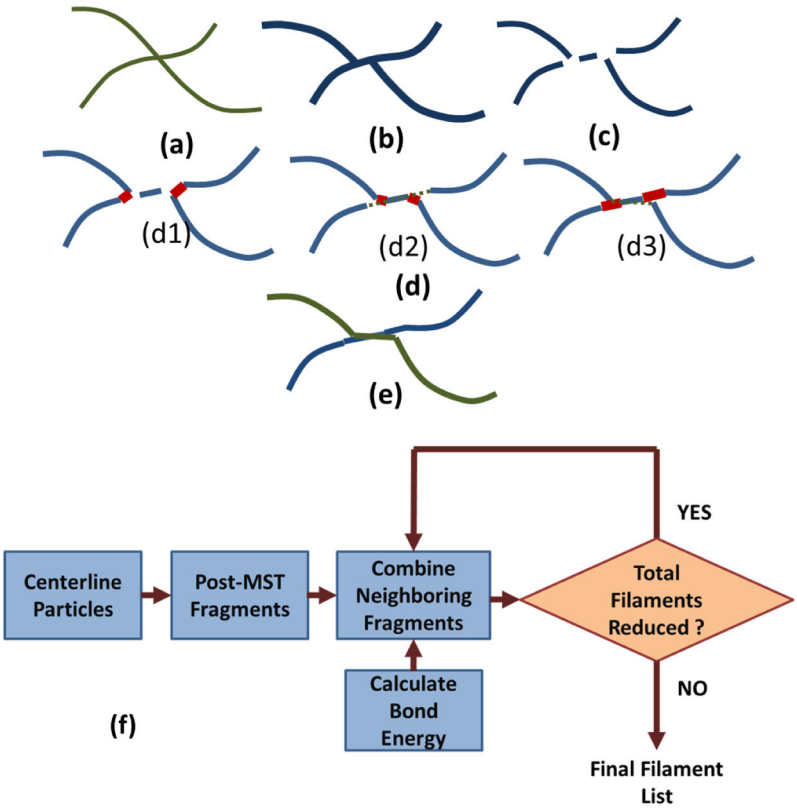

Fig. 2.

Schematic of combination of longer filaments from shorter filament segments. (a) Suppose we have two filaments intersecting as shown in green in the original confocal image. (b) after the reverse diffusion step in section II-A and the MST generation step in the beginning of section II-B1, the corresponding tree estimate for (a) is shown in blue. (c) The bifurcation nodes in the tree in (b) are omitted to give shorter non-intersecting fragments as outlined in II-B1. (d) d1, d2 and d3 show three out of the numerous ways of joining neighboring fragments to form longer filaments, where red thick lines and green dotted lines show the corresponding joining of neighboring filaments. For each of these joining, the combined curvature energy of the longer filament increases with respect to the sum of the curvature energies of the shorter constituent filaments; this is balanced by the bond energy b. (e) d3 is chosen to be an optimum configuration of joining the fragments in this particular step of the iteration since it minimizes the overall curvature energy of the configuration. The two resultant longer filaments are shown in blue and green. This iterative combination is now continued with the longer filaments considered as starting constituent filaments. The idea of iteratively combining neighboring filament fragments to gradually give rise to longer and complete filaments is shown in a flowchart in (f).

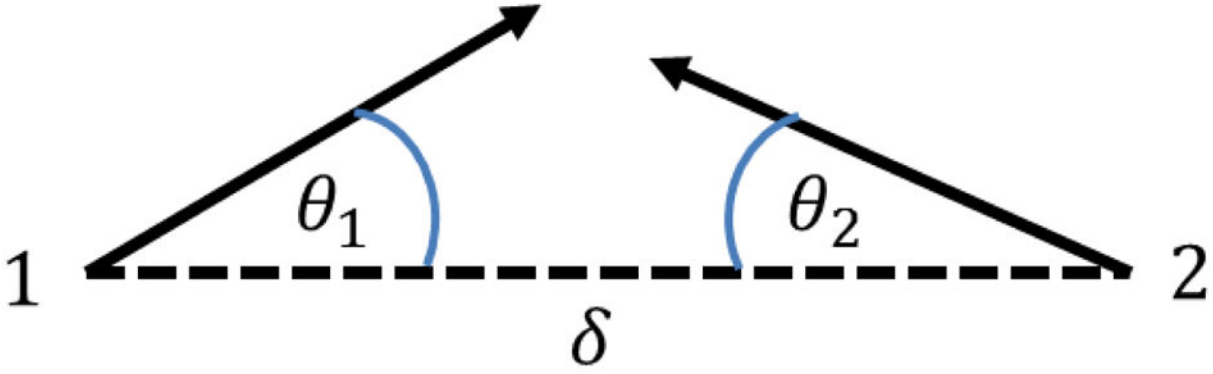

Fig. 3.

The schematic for calculating the bond energy b. 1 and 2 denote two endpoints of two different filament segments separated by a distance δ, and θ_1 and θ_2 are orientations of the tangents at the respective endpoints.

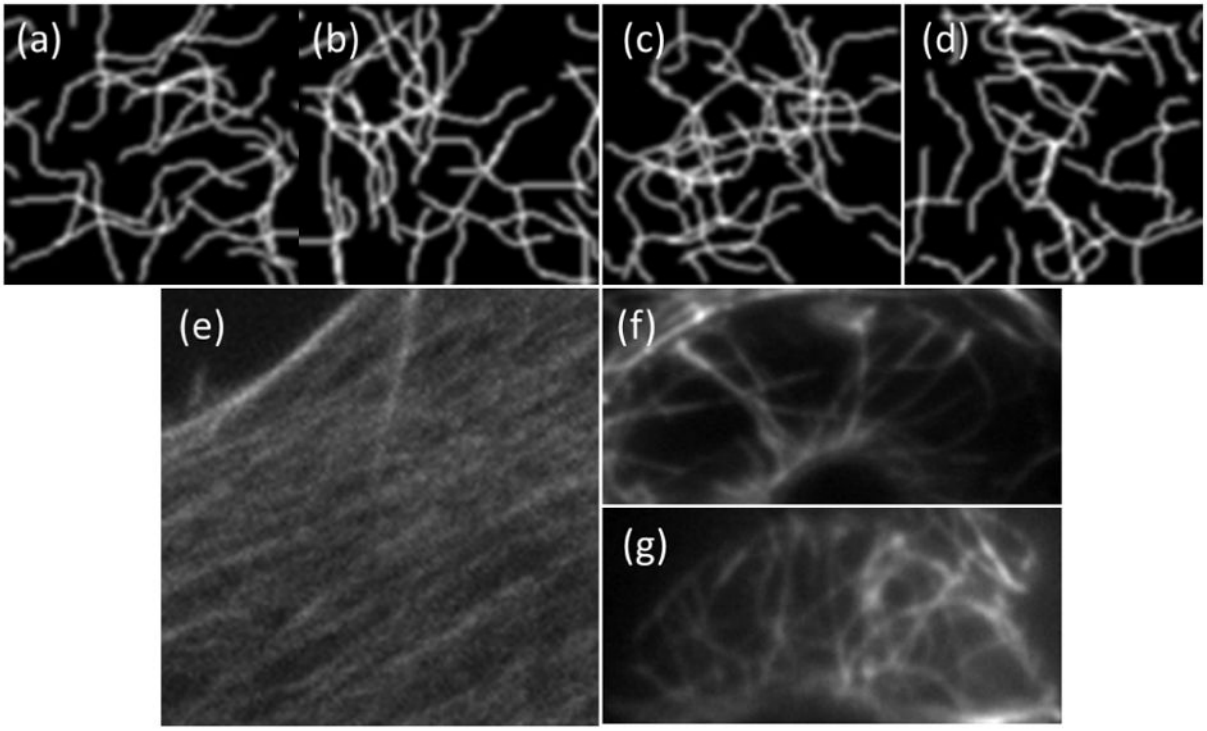

Fig. 4.

(a), (b), (c) and (d) show four images from a simulated database of artificial filaments with filament counts of 33, 34, 35 and 36 respectively. (e) shows a real image of HeLa cell with rhodamine phalloidin labeled F-actin [28] and (f), (g) shows two maximum intensity projection views of plant stomata guard cells stained for actin microfilaments [7].

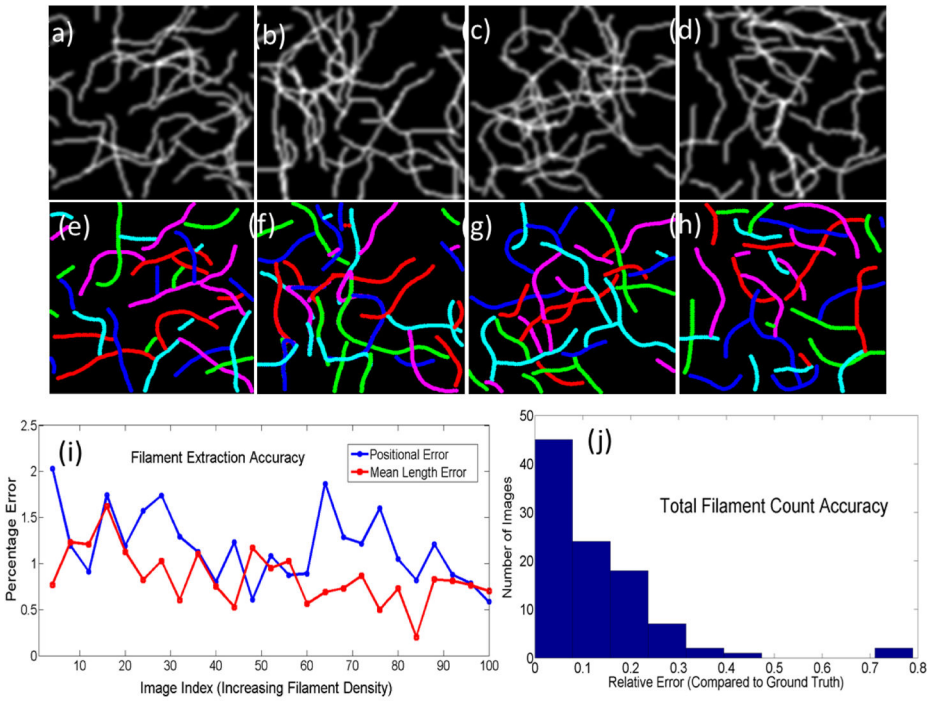

Fig. 5.

(a), (b), (c) and (d) show the same four simulated images from Fig. 4, and (e), (f), (g) and (h) shows the corresponding filament extracted images. Different colors in (e)–(h) denote different filaments, although colors are not unique. (i) shows the filament extraction accuracy in percentage error. The blue plot shows the positional error which is estimate of an average error of matching the closest filament pairs, one filament coming from the original image and the other coming from the extracted filament set (see text for details). The red plot shows the percentage error in estimation of mean length of the filament images (see text for details) (j) shows the histogram of relative error in determining the total filament count in an image (see text for details of calculation of relative errors).

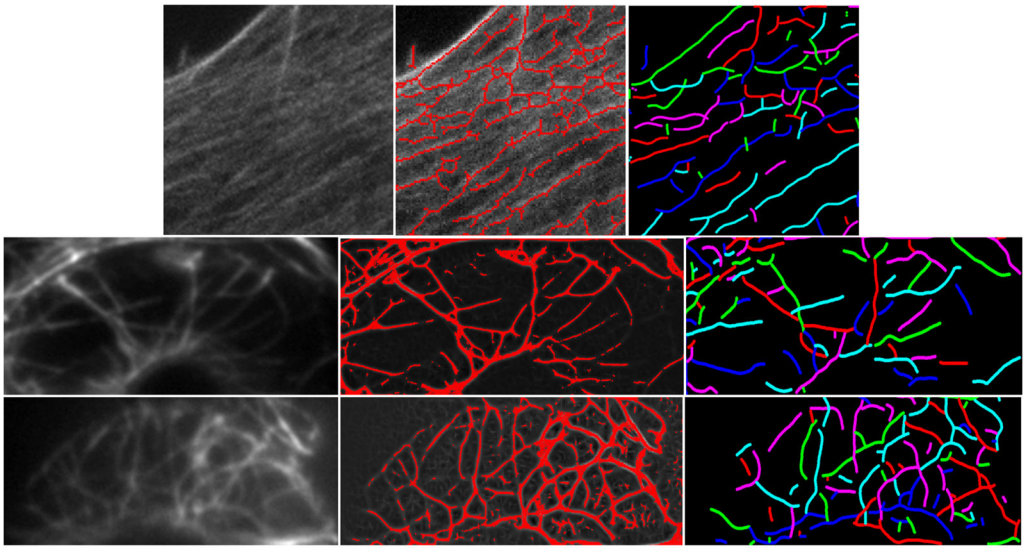

Fig. 6.

The first column shows the four raw images of actin stained HeLa (first row) and plant stomata guard (second and third rows) cells. The second column shows the localized centerlines of the actin filaments after the reverse diffusion step in section II-A of the corresponding images (red pixels overlaid on the raw images). The third column shows the corresponding filament extracted images, where individual filaments are shown in different colors.

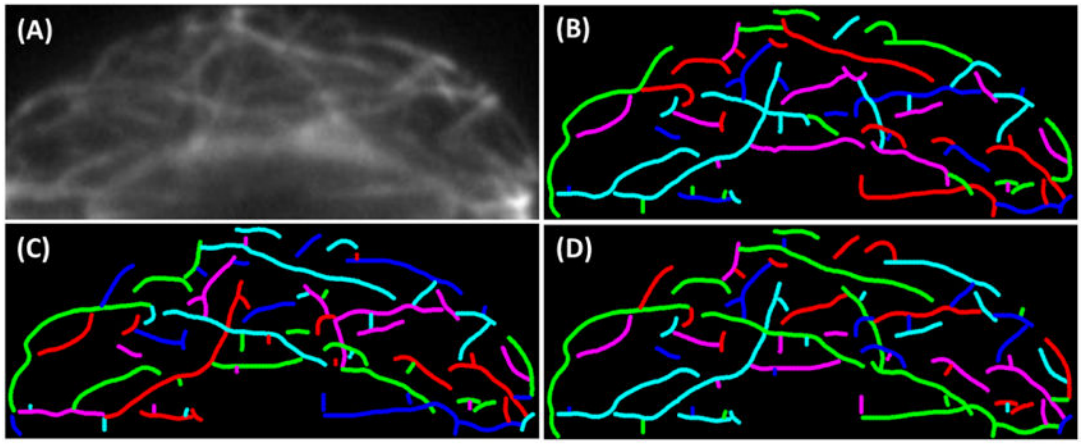

Fig. 7.

Filament extractions for the actin filaments in the *Arabidopsis* guard cells similar to Fig. 6 (second and third row). (A) The original actin microfilament maximum intensity projection image in a sample guard cell. Keeping the same values for parameters in the filament localization step, we have used the maximum segment gap δ of 4, 6 and 8 pixels and maximum orientation mismatch θ of 15, 20 and 25 degrees respectively in (B), (C) and (D). It can be seen there are negligible differences in extracted filaments in (B), (C) and (D).

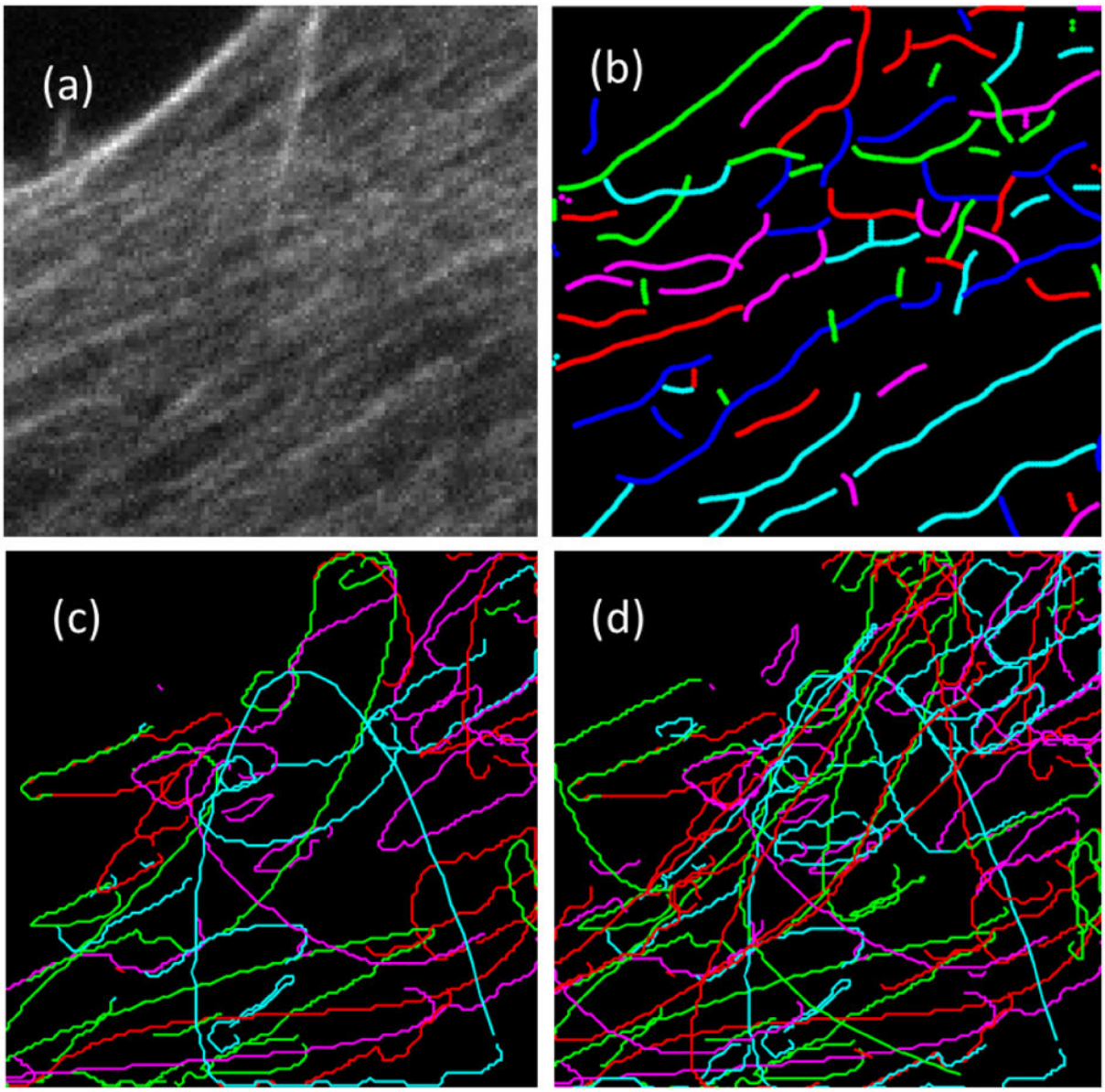

Fig. 8.

A simple experiment to demonstrate the inapplicability of seed following algorithms such as [21] to our microscopy images. (a) shows the original image stained for actin and (b) shows the result of actin filament exraction by our algorithm. (c) and (d) show the performance of the 2D version of [21] with two different maximum parameter extraction settings. See text for more details.

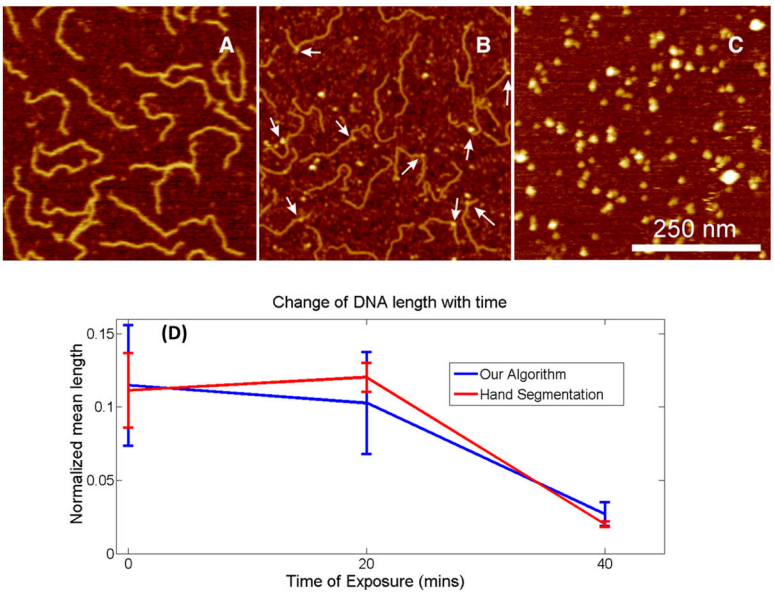

Fig. 9.

AFM images of DNA fragments deposited on mica as shown in Fig. 1 in [16]. (A), (B) and (C) refer to the DNA panels prior to, after 20 mins ad after 40 mins respectively of UV radiation. The bottom plot shows the normalized mean length of the DNA fragments in the above panels, plotted as a function of time of exposure, estimated with the help of our filament extraction algorithm. The red plot shows that performed by a human, and the blue plot shows the same estimation performed by our algorithm.